

Máster en Programación avanzada en Python para Big Data, Hacking y Machine Learning

Programación Python para Machine Learning

LECCIÓN 08

Lección 8: Redes Neuronales Artificiales.

ÍNDICE

- ✓ Introducción
- ✓ Objetivos
- ✓ Principios teóricos y conceptos de las Redes Neuronales
- ✓ Implementación de una Red Neuronal
- ✓ Consideraciones a tener en cuenta.
- ✓ Conclusiones

INTRODUCCIÓN

- ✓ Inspiración del modelo: sistema nervioso.
- ✓ Capacidad de identificar patrones.
- ✓ Redes Neuronales Artificiales: modelo de Machine Learning supervisado no lineal.

OBJETIVOS

Al finalizar esta lección serás capaz de:

- 1 Conocer los principios en los que se basan las Redes Neuronales Artificiales.
- 2 Entender los elementos clave que conforman una Red Neuronal.
- 3 Implementar un modelo de Red Neuronal en Python para resolver problemas de clasificación y regresión.
- 4 Identificar los aspectos a tener en cuenta para mejorar el rendimiento de una Red Neuronal.

REDES NEURONALES ARTIFICIALES

Redes Neuronales: modelos de ML que realizan operaciones matemáticas siguiendo una estructura de red.

Capas → Nodos (Perceptrones).

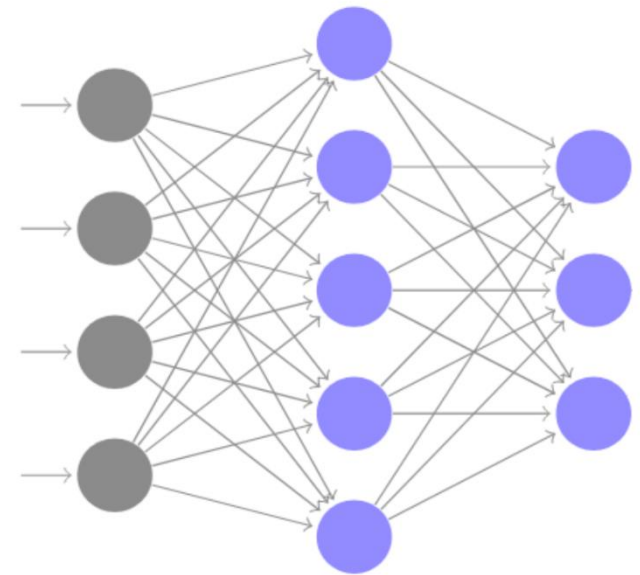
También llamadas MLP.

Capa de entrada.

Capa(s) oculta(s).

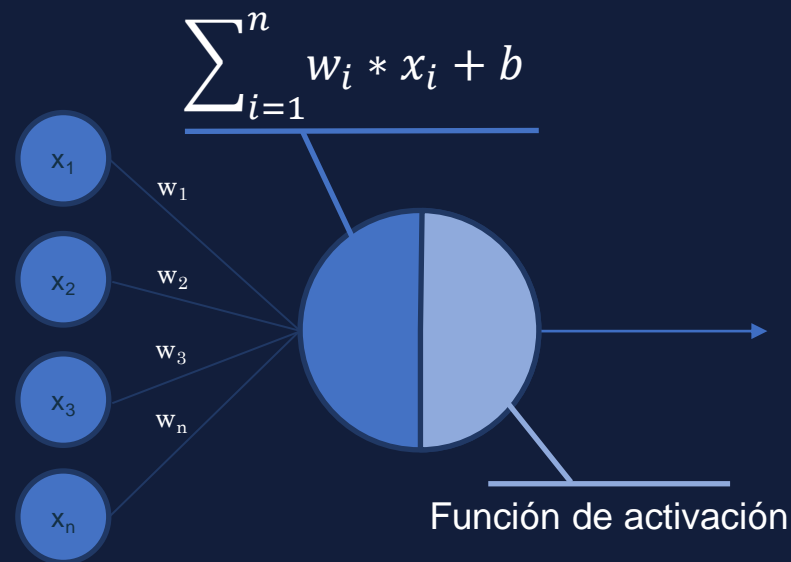
Capa de salida.

Conexiones → Pesos.

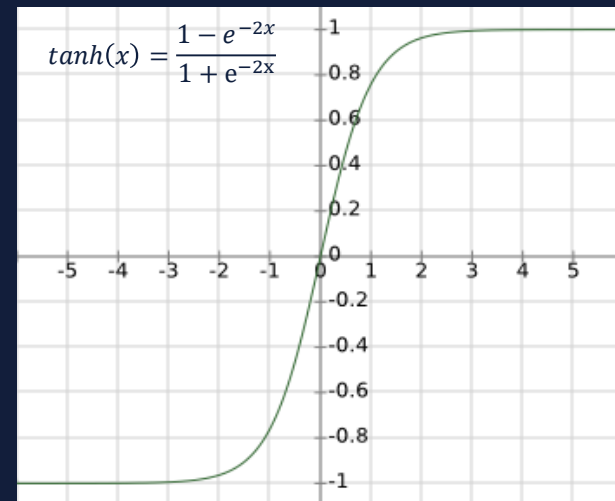
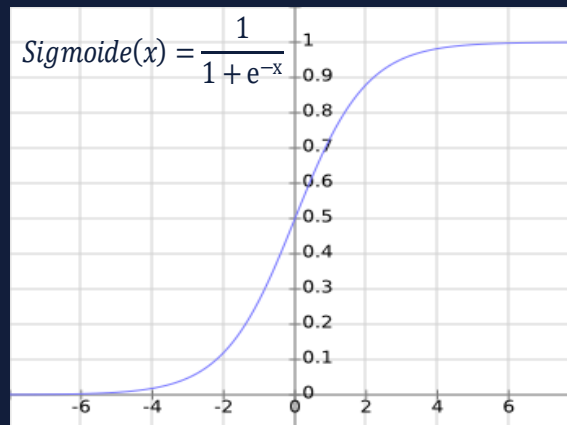
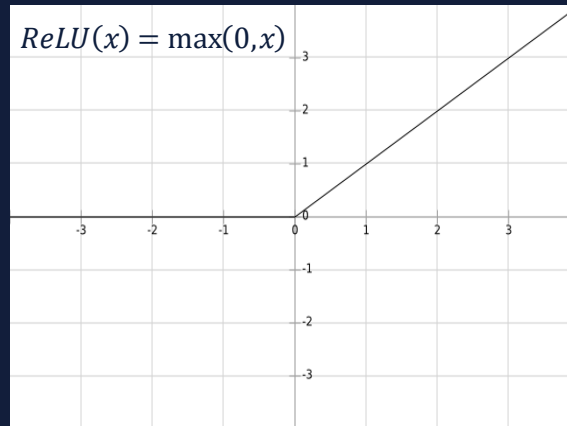


REDES NEURONALES ARTIFICIALES

Perceptrón: unidad funcional de una red neuronal.



REDES NEURONALES ARTIFICIALES



Entrenamiento de una Red Neuronal

El proceso de **entrenamiento** de una red neuronal es el siguiente:

1. Inicializar una red con valores para pesos y bías de modo aleatorio.
2. Cada patrón de entrenamiento es utilizado para calcular el error que comete la red en la salida.
3. Determinar el grado con el que cada peso y bías a influenciado en el error de la predicción.
4. Alterar los pesos de la red de modo proporcional a su responsabilidad en el error.

SGD + Backpropagation

Función de pérdida: RMSE (regresión) ó LogLoss (clasificación)

Entrenamiento de una Red Neuronal

El rendimiento de las Redes Neuronales está muy condicionado al ajuste de sus parámetros:

1. La topología de la red.
2. La tasa de entrenamiento.
3. El número máximo de épocas.
4. El momento, solo en caso del solver 'sgd'.

La función de pérdida que implementan es no convexa, existe más de un mínimo local. Esto implica que no siempre haya una estabilidad en su rendimiento.

MUCHAS GRACIAS POR SU ATENCIÓN



jperez@grupomainjobs.com



Javier Pérez Rodríguez
www.linkedin.com/in/perezxavi



twitter.com/eiposgrados



facebook.com/eiposgrados



instagram.com/eiposgrados